

**Examen - 16 de julio de 2016**

- Duración de esta etapa: 3 Hs.
- No se podrá utilizar ningún tipo de material (apuntes, libro, calculadora, etc). Apague su celular.
- **Sólo** se contestarán preguntas sobre interpretación de la letra hasta 30 minutos antes de la finalización del mismo.
- Las partes no legibles del parcial se considerarán no escritas
- En la primer hoja a entregar ponga con letra clara, en el ángulo superior derecho, salón en el cual desarrolló la prueba, su nombre, número de cédula de identidad y cantidad de hojas -en ese orden-; las demás hojas es suficiente con nombre, número de cédula y número de página.

Para la resolución de los diferentes ejercicios **solamente** podrá utilizar las siguientes funciones brindadas por **Matlab**:

- length() y size()
- mod() y rem()
- floor(), ceil() y round()
- zeros() y ones()

<b>Problema 1</b>	15 ptos (3,3,3,3,3)	
-------------------	---------------------	--

- Calcule la expresión decimal del siguiente número binario puro: 101011.
- Calcule la expresión en hexadecimal de  $1011011010111_2$  (en base 2).
- Represente en complemento a 1 con 6 bits, del número 13.
- Determine la expresión decimal que representa la tira 0 00100010 010100000000000000000000 en punto flotante simple precisión.
- Determine la representación en el sistema de punto flotante simple precisión de 1,25.

---

**Nota:** Justificar todas las respuestas.

---

<b>Problema 2</b>	20 (10,10) ptos	
-------------------	-----------------	--

- Implementar en Matlab la función **recursiva** *EsPar* la cual recibe como parámetro un vector columna de enteros y devuelve 1 si la suma de los elementos es par, 0 en caso contrario.
- Implementar en Matlab la función **iterativa** *AlgunPar* la cual recibe como parámetro una matriz de enteros y devuelve 1 si a lo sumo una de las columnas de la misma cumple que la suma de todos sus elementos es par, 0 en caso contrario.

---

**Nota:** Para la parte b) se puede utilizar la función implementada en la parte a).

---

<b>Problema 3</b>	25 ptos (10,15)	
-------------------	-----------------	--

- Implementar en Matlab la función **iterativa** *IndMayorIt* la cual recibe como parámetro un vector y devuelve el índice donde se encuentra el máximo.
- Implementar en Matlab la función **recursiva** *IndMayorRec* la cual recibe como parámetro un vector y devuelve el índice donde se encuentra el máximo. Puede utilizar una función con cabezal.

---

**Nota:** Asumir que solo existe un máximo en el vector.

---

<b>Problema 4</b>	20 (10,10) pts	
-------------------	----------------	--

a) Implementar en Matlab la función **iterativa** *DimMin* la cual recibe una matriz dispersa en formato elemental y devuelve un vector con las dimensiones mínimas (cantidad de filas y cantidad de columnas) que debe tener esa matriz.

---

**Nota:** Los elementos de la matriz en formato elemental no están ordenados.

---

b) Implementar en Matlab la función **iterativa** *ExcedeX* la cual recibe una matriz dispersa en formato elemental y un entero  $x$  y devuelve 1 si la suma de los valores de la matriz en las posiciones pares es mayor que el valor  $x$  y 0 en otro caso.

---

**Nota:** Un valor se encuentra en una posición par de la matriz si ambas coordenadas de ese valor son pares.

---

<b>Problema 5</b>	20 (10,10) pts	
-------------------	----------------	--

La relación de recurrencia del polinomio de Legendre se escribe como:

$$P_0(z) = 1$$

$$P_1(z) = z$$

$$P_n(z) = \frac{(2n-1)}{n} z P_{n-1}(z) - \frac{(n-1)}{n} P_{n-2}(z)$$

a) Implementar en Matlab la función **iterativa** *LegIt* la cual recibe  $n$  y  $z$ , y devuelve la evaluación del polinomio  $P_n$  en  $z$ .

b) Implemente una versión **recursiva** *LegRec* de la función de la parte a).